

## CLAIMS

What is claimed is:

1. A method for providing a high speed parallel apply in asynchronous data replication in a database system, the database system including a source node and a target node, comprising:

5 (a) examining a transaction message in a receive queue at a target node, wherein the transaction message comprises information concerning at least one row change to a table copy at the source node;

10 (b) determining if the transaction message has any dependencies on at least one preceding non-completed transaction message;

15 (d) holding the transaction message, if the transaction message has dependencies on the preceding non-completed transaction message;

(e) placing the transaction message on a work queue, if the transaction message has no dependencies on any preceding non-completed transaction message;

(f) examining a done queue for completed transaction messages to determine if the completed transaction messages remove the dependencies of the held transaction message; and

15 (g) placing the held transaction message on the work queue, if the completed transaction messages remove the dependencies of the held transaction message.

2. The method of claim 1, further comprising:

(h) examining a plurality of transaction messages on the work queue by a

plurality of agent threads;

(i) applying in parallel row changes in each of the plurality of transaction

messages by each of the plurality of agent threads;

(j) updating a control table to indicate completion of the application of each of the plurality of transaction messages; and

(k) placing each completed transaction message on the done queue.

10 3. The method of claim 2, further comprising:

(l) examining each completed transaction message on the done queue;

(m) determining if the completion of the completed transaction message clears the dependencies of any of the held transaction messages dependent upon the completed transaction message; and

15 (n) placing any of the held transaction messages onto the work queue, if the dependencies of the held transaction message have been cleared.

4. The method of claim 1, wherein for each row change in the transaction

message, the determining (b) comprises:

20 (b1) determining that the row change in the transaction message is an insert or a key update type of change;

(b2) comparing a new replication key value in the row change in the transaction message to an old replication key value of a row change in the preceding transaction message; and

(b3) determining that the transaction message has dependencies if the new replication key value in the row change in the transaction message is the same as the old replication key value in the row change in the preceding transaction message.

5. The method of claim 4, wherein the comparing (b2) comprises:

(b2i) comparing a hash value of the new replication key value in the row change in the transaction message to a hash value of the old replication key value in the row change in the preceding transaction message.

6. The method of claim 1, wherein for each row change in the transaction message, the determining (b) comprises:

15 (b1) determining that the row change in the transaction message is a delete or a key update type of change;

(b2) comparing an old replication key value in the row change in the transaction message to a new replication key value in a row change in the preceding transaction message; and

20 (b3) determining that the transaction message has dependencies if the old replication key value in the row change in the transaction message is the same as the new

replication key value in the row change in the preceding transaction message.

7. The method of claim 6, wherein the comparing (b2) comprises:

5 (b2i) comparing a hash value of the old replication key value in the row change in the transaction message to a hash value of the new replication key value in the row change in the preceding transaction message.

8. The method of claim 1, wherein for each row change in the transaction message, the determining (b) comprises:

10 (b1) determining that the row change in the transaction message is an update type of change;

(b2) comparing a new replication key value in the row change in the transaction message to a new replication key value in a row change in the preceding transaction message; and

15 (b3) determining that the transaction message has dependencies if the new replication key value in the row change in the transaction message is the same as the new replication key value in the row change in the preceding transaction message.

9. The method of claim 8, wherein the comparing (b2) comprises:

20 (b2i) comparing a hash value of the new replication key value in the row change in the transaction message to a hash value of the new replication key value in the row change in

the preceding transaction message.

10. The method of claim 2, further comprising:

(l) removing the completed transaction message from the receive queue.

5

11. The method of claim 10, wherein the removing (l) comprises:

(l1) deleting the completed transaction message from the receive queue as part of

a two-phase commit synchronization with the application of the completed transaction message.

10

12. The method of claim 10, wherein the removing (l) comprises:

(l1) obtaining at least one entry in a control table at the target node indicating that the completed transaction message has been completed; and

(l2) deleting the completed transaction message from the receive queue.

15

13. The method of claim 12, further comprising:

(l3) removing the at least one entry from the control table.

20

14. A computer readable medium with program instructions for providing a high speed parallel apply of in asynchronous data replication in a database system, the database system including a source node and a target node, comprising:

(a) examining a transaction message in a receive queue at a target node, wherein the transaction message comprises information concerning at least one row change to a table copy at the source node;

5 (b) determining if the transaction message has any dependencies on at least one preceding non-completed transaction message;

(d) holding the transaction message, if the transaction message has dependencies on the preceding non-completed transaction message;

(e) placing the transaction message on a work queue, if the transaction message has no dependencies on any preceding non-completed transaction message;

10 (f) examining a done queue for completed transaction messages to determine if the completed transaction messages remove the dependencies of the held transaction message; and

(g) placing the held transaction message on the work queue, if the completed transaction messages remove the dependencies of the held transaction message.

15

15. The medium of claim 14, further comprising:

(h) examining a plurality of transaction messages on the work queue by a plurality of agent threads;

(i) applying in parallel row changes in each of the plurality of transaction messages by each of the plurality of agent threads;

(j) updating a control table to indicate completion of the application of each of

the plurality of transaction messages; and

(k) placing each completed transaction message on the done queue.

16. The medium of claim 15, further comprising:

5 (l) examining each completed transaction message on the done queue;

(m) determining if the completion of the completed transaction message clears the dependencies of any of the held transaction messages dependent upon the completed transaction message; and

10 (n) placing any of the held transaction messages onto the work queue, if the dependencies of the held transaction message have been cleared.

17. The medium of claim 14, wherein for each row change in the transaction message, the determining (b) comprises:

15 (b1) determining that the row change in the transaction message is an insert or a key update type of change;

(b2) comparing a new replication key value in the row change in the transaction message to an old replication key value of a row change in the preceding transaction message; and

20 (b3) determining that the transaction message has dependencies if the new replication key value in the row change in the transaction message is the same as the old replication key value in the row change in the preceding transaction message.

18. The medium of claim 17, wherein the comparing (b2) comprises:

(b2i) comparing a hash value of the new replication key value in the row change in the transaction message to a hash value of the old replication key value in the row change in the preceding transaction message.

5

19. The medium of claim 14, wherein for each row change in the transaction message, the determining (b) comprises:

(b1) determining that the row change in the transaction message is a delete or a key update type of change;

10 (b2) comparing an old replication key value in the row change in the transaction message to a new replication key value in a row change in the preceding transaction message; and

15 (b3) determining that the transaction message has dependencies if the old replication key value in the row change in the transaction message is the same as the new replication key value in the row change in the preceding transaction message.

20. The medium of claim 19, wherein the comparing (b2) comprises:

20 (b2i) comparing a hash value of the old replication key value in the row change in the transaction message to a hash value of the new replication key value in the row change in the preceding transaction message.

21. The medium of claim 14, wherein for each row change in the transaction message, the determining (b) comprises:

(b1) determining that the row change in the transaction message is an update type of change;

5 (b2) comparing a new replication key value in the row change in the transaction message to a new replication key value in a row change in the preceding transaction message; and

10 (b3) determining that the transaction message has dependencies if the new replication key value in the row change in the transaction message is the same as the new replication key value in the row change in the preceding transaction message.

22. The medium of claim 21, wherein the comparing (b2) comprises:

15 (b2i) comparing a hash value of the new replication key value in the row change in the transaction message to a hash value of the new replication key value in the row change in the preceding transaction message.

23. The medium of claim 15, further comprising:

(l) removing the completed transaction message from the receive queue.

20 24. The medium of claim 23, wherein the removing (l) comprises:

(l1) deleting the completed transaction message from the receive queue as part of

a two-phase commit synchronization with the application of the completed transaction message.

25. The medium of claim 23, wherein the removing (l) comprises:

5 (l1) obtaining at least one entry in a control table at the target node indicating that

the completed transaction message has been completed; and

(l2) deleting the completed transaction message from the receive queue.

26. The medium of claim 25, further comprising:

10 (l3) removing the at least one entry from the control table.

27. A system, comprising:

a source node, wherein the source node sends a transaction message concerning a committed transaction completed at a source table copy to a target node to asynchronously replicate the transaction; and

15 the target node, wherein the target node comprises a receive queue, a browser thread, a work queue, a done queue, at least one agent thread, and a target table copy,

wherein the transaction message concerning the transaction is received on the receive queue,

20 wherein the browser thread examines the transaction message to determine if the transaction message has any dependencies on at least one preceding non-complete

transaction message,

wherein the transaction message is held by the browser thread if the transaction message has dependencies on the preceding non-completed transaction message,

wherein the transaction message is placed on the work queue if the transaction message has no dependencies on any preceding non-completed transaction message,

wherein the done queue is examined for completed transaction messages to determine if the completed transaction messages remove the dependencies of the held transaction message,

wherein the held transaction message is placed onto the done queue if the completed transaction messages remove the dependencies of the held transaction message.